ChinaPay NetPayClient API
Merchant Manual

Version 2.5.18

Revision History

Date	Version	Description	Author
2012-02-29	2.5.18	Add parameter BgRetUrl description in chapter 4.3	Hu.jianlin
2010-2-18	2.5.17	Add function definition of PHP in Chapter 3.5	Zhou.yin/Ye.qingqing
2010-1-16	2.5.16	Change response code in the chapter of Query from "ResponseCode" to "ResponeseCode"	Ye.qingqing
2010-11-15	2.5.15	Change the value of return URLs in Payment and Refund	Ye.qingqing
2010-09-02	2.5.14	Modifying query version 20060831 digital signature string field ` OrderId` to `OrdId`, modifying java function reference link `sign` to `Sign`	li.guolin
2010-08-24	2.5.13	NetPayClient_Merchant Operational Manual domestic version.	li.guolin

Contents

1. Backgrounds

ChinaPay commits herself to the development of e-payment service in China.

The thriving economic development of China has brought us not only the e-commerce based on leading technology but also the concept of e-finance services. Today, both traditional businesses and general consumers are in greater and higher need for best-of-breed financial services. For businesses, the presence of e-commerce and other brand-new business models impel them to request for high-grade electronization and automatization of the financial services. While for consumers, their needs for personal online financial services are continuously increasing as a result of life quality improvement. ChinaPay, a new-born financial service representative, embraced the world leading hi-technology into its professional financial services provided to businesses and consumers in a customer-oriented manner. It is the aim to enable the whole society to enjoy the e-payment service, to reduce costs, to enhance social working efficiency, and to improve the living standard accordingly.

ChinaPay highly appreciates the great support from PBOC, local commercial banks and SIIC in our process of development. We provide services including enterprise-oriented e-payment resolutions and e-finance services focused on individual consumers. Pursuing for breakthroughs and innovations, the products and resolutions developed by ChinaPay integrate the Chinese-characterized professional financial services with the modern hi-technology to meet the market demand and are also clinging to the principle of professional, safe and easy to use. Employed by hundreds of businesses and more than a hundred thousand consumers, ChinaPay's e-payment services have been proved to be a comprehensive and professional support for businesses' e-commerce as well as a secure and fast individual financial service for general consumers.

ChinaPay is working to establish a nation-wide and multi-industry financial e-payment network system. By cooperation with commercial banks and hi-tech enterprises, we will be able to provide all-needs-met e-payment services to fulfill a common prosperity for all parties involved.

2. Introduction

NetPayClient is an API Lib installed on the client side of ChinaPay's authorized merchants. It integrates with merchants' online system to provide secure online payment method between customers, merchants and banks. Its main functions can be described as follows,

- To realize bankcard online payments
- To refund successful payment transactions
- To inquire existing onlinepayment transactions

2.1. Object

The main object of this manual is to help the member merchants of ChinaPay use the payment service interface- NetPayClient.

The document has four parts：

Summary: Introduces some basic concepts in the file, NetPayClient and its usage

Payment：Introduces required data field and content of payment interface

Refund： Introduces required data field and content of refund interface

Query： Introduces usage of transaction inquiry interface

3. Summary

3.1. Why read this?

The purpose of this document is to help ChinaPay's merchants to understand ChinaPay's online transaction flows, NetPayClient installation and usage, which will facilitate the system integration process with merchant's online system.

This manual is intended as a guide for system developers, administration and maintenance personnel of ChinaPay's merchants. They are required to have basic knowledge about the followings,

- Knowledge of operation systems, such as Microsoft Windows/NT、Windows9x、Windows 2000、HP-UX、AIX、SUN Solaris、Linux、BSD.
- Knowledge of website configuration and web page development based on any of the operation systems listed above.
- Development skills of standard CGI（Common Gateway Interface）, ASP（Active Server Pages）, ISAPI, PHP or JAVA.
- Basic understandings of information security.

3.2. System environment

Merchants can choose the proper development and run-time environment according to the practice. Currently, ChinaPay is able to support the following platforms and development tools,

- NetPayClient for Java （in JAR，for Java development）
- NetPayClient for Win32 （in DLL，for Windows system development）
- NetPayClient for C (in.so or.a，for UNIX LINUX FreeBSD etc.development)
- NetPayClient for C# （in DLL，for Windows system development）

3.3. Keywords

3.3.1. E-commerce

Electronic commerce refers to those activities, including business activities, financial activities and other related service activities, carried out worldwide by means of electronic information exchange via networks.

3.3.2. Networking trading

Refers to the online trading on the basis of networks, which makes the whole process in automation or part automation from browsing, negotiation, contract, delivery to payment by connecting all the relevant parties together, such as business, custom, logistics, commodity inspection, with the help of digital technologies. Network trading consists of five parts, namely information share, order making, payment, execution and also service and support.

3.3.3. B2C(Business to Consumer)

Refers to the e-commerce activity between business and customer. This approach enables customers to take part into the economic activities via Internet directly. Nowadays, there are a great deal of virtual shops and virtual businesses on the Internet to provide services of product selling.

3.3.4. B2B(Business to Business)

Refers to the e-commerce model between business and business. The businesses are able to use the Internet or the other types of network to seek the most appropriate partner for each transaction, and also to complete it from order making to settlement.

3.3.5. Merchant

The shops and businesses those are capable to perform e-commerce business.

3.3.6. Bankcard

The payment media issued by commercial banks, including saving cards, credit cards, debit cards, enterprise cards, co-brand cards and virtual e-cards etc.

3.3.7. Cardholder

The owner of the bankcard

3.3.8. Online Payment Password

A numeric string used by the issuing bank to identify the "online cardholder". Please check with the particular issuing bank for how to apply for it as well as how to use it.

3.3.9. Issuer

The commercial bank that issues the bankcards

3.3.10. Accounting Bank

Refers to the commercial bank that the personal account or the corporate account open in.

3.3.11. SSL(Secure Socket Layer)

SSL is a transaction security standard that provides functions of data encryption, server authentication, and message integrity. SSL helps to improve the safety of client/server Internet communication.

3.4. Basic concepts

With the development of computers and Internet, more and more businesses, merchants and financial institutions have started to migrate their traditional business to the Internet gradually so as to reduce costs, increase efficiency, explore new market areas and improve service quality.

Internet is no longer a simple information exchange platform; instead it has grown up into a global business network. Consequentially, network security is drawing more and more attention nowadays. What mean by network security is to solve the following problems using all kinds of available technologies,

Privacy – the information is protected from being sniffed by the third party during transmission.

Integration – the receiver is enabled to detect whether the information has been

changed during transmission or not.

Authenticity – the receiver is able to identify the sender's identity.

Undeniability – the sender shall not be able to deny his sending activity as long as the other party receives the effective information sent out.

Modern cryptography gives satisfying answers to these questions. As known, encryption algorithms are responsible to solve the information privacy concern, message-digest algorithms make it possible for integration checking, digital signature enables the function of authentication and also avoids the deny after the event. Modern cryptography comprises two systems; they are public-key cryptography and secret-key cryptography. This chapter will explain some basic concepts briefly.

3.4.1. Public-key Cryptography

Public-key cryptography is also called asymmetrical cryptography, which utilizes two keys, one for message encryption, and one for message decryption. This algorithm is based on a mathematical relation between those two keys, so that the message encrypted using either of the keys could only be decrypted using another key in the pair. Each user owns a pair of keys, one is called private key that the user is responsible to keep it private, and the other called public key is made widely available so that anyone can possess it. Data encrypted with a public key can be decrypted only with the corresponding private key, and vice versa. Currently, the most well known public key cryptography is called RSA algorithm as it is invented by Rivest, Shamir and Adleman.

Figure 1 RSA Algorithm

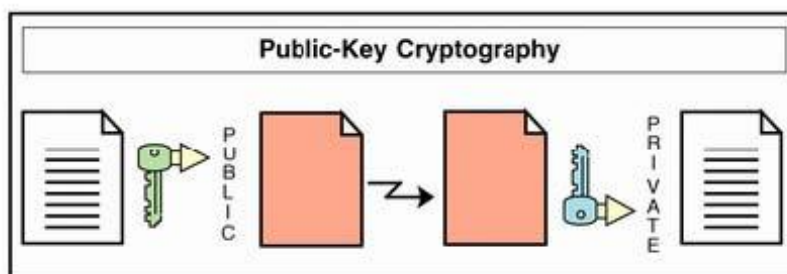### 3.4.2. Secret-Key Cryptography

Secret-key cryptography, known as symmetrical cryptography, employs only one key for both encryption and decryption. A pair of users shares one secret key and is responsible to keep it confidential between them. The message encrypted by the secret key could only be decrypted using the same key. The most prevalent algorithms are DES(Data Encryption Standard), 3DES etc.
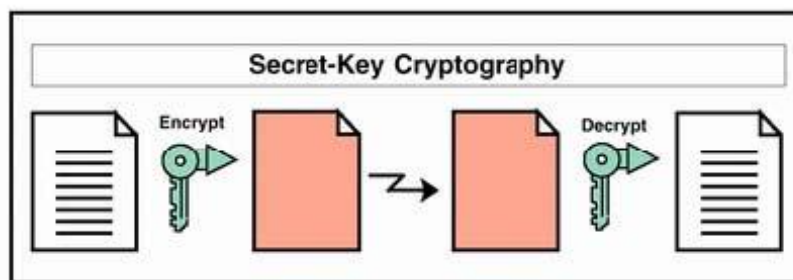
Figure 2. DES Algorithm

3.4.3. Message-digest Algorithm

Message-digest algorithm is a method taking a message of arbitrary length and producing a message digest of fixed-length. The digests are sometimes called the "digital fingerprints" of data. A sound message-digest algorithm should have the following features,

It is almost impossible to find two such different byte streams that give the same digest.

If given a digest value, it is hard to invert, that is to say, it is computationally infeasible to find the input message.

Before the sender sends a message, a message digest is applied to the message and attached to it. Upon receiving, the recipient takes the original message and runs it through the same message-digest algorithm to create his own local message digest value. The recipient then compares the sender provided message digest with the locally recovered message digest to check whether they are identical. If they are indeed identical, this fact assures that there has been no change or tampering during transmission. The widely used message-digest algorithm includes MD2 and MD5. Please check document RFC1319 and RFC1321 for the usage of these two algorithms.

Figure 3. Message Digest

3.4.4. Digital Signature

A digital signature is an application that combines the asymmetrical cryptography with message-digest algorithms. By using this, a digital signature authenticates the integrity of

the signed data and the identity of the signatory. The following example explains the digital signature process in details.
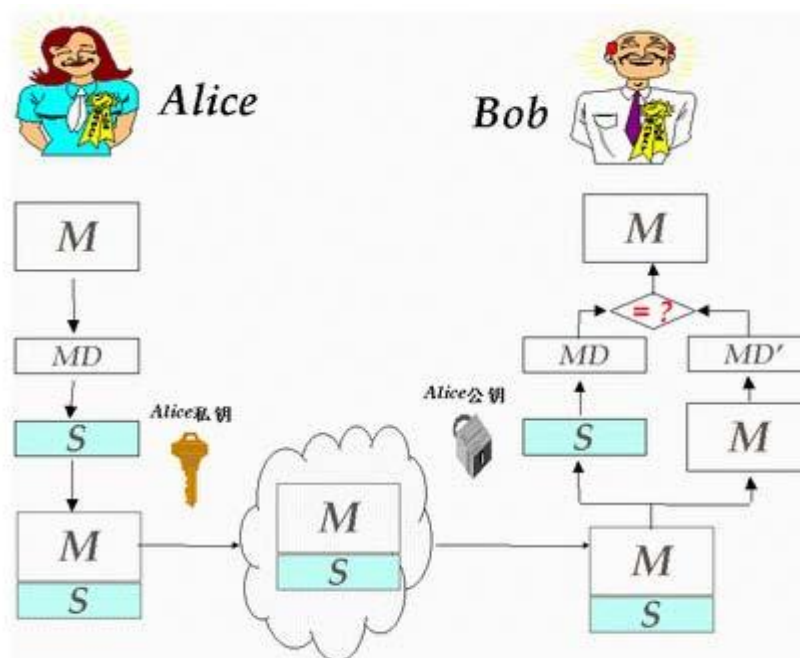
Figure 4. Digital Signature

3.4.5. Digital Envelope

Alice has a contract M.

Alice uses message-digest algorithm to calculate the message digest of the contract M, say MD.

Alice uses asymmetric algorithm with her private key to sign the message digest MD, and the result S is the so-called digital signature.

Alice attaches the contract M with the signature S, and then sends it to the recipient, Bob, via the network.

Bob receives the contract M with the signature S sent by Alice.

Bob uses Alice's public key to decrypt the signature S so as to obtain the message digest MD which was prepared by Alice before sending.

Bob takes the originally unencrypted contract M and runs it through the same message-digest algorithm to create his own local message digest MD'.

Bob compares MD and MD' to check if they are identical.

If they are indeed identical, then the message is verified. The recipient knows that the signatory has sent the message because only the sender's public key will work. The message was signed by no one other than himself, and that the message has not been modified since he signed it.

3.5. Installing and deploying NetPayClient

3.5.1. Function Definitions

NetPayClient is an API Lib installed on the client side of ChinaPay's authorized merchants. It integrates with the online system to provide secure online payment service between customers, merchants and banks. Its main functions can be described as follows:

. To generate digital signature for orders key information

. To verify the response of transaction results

. To generate digital signature for a digital string

. To verify digital signature of a digital string

3.5.2.   NetPayClient for Java Function Description

3.5.2.1. File list

Name	Path	Usage
netpayclinet.jar	Installed in the corresponding path according to the need of project	To provide the method of deploying digital signature
MerPrk.key	Unspecified, but needed to deploy function appointed file path and file name	Merchant's private key
PgPubk.key	Unspecified, but needed to deploy function appointed file path and file name	ChinaPay's public key

3.5.2.2. Functions Description

1)   Function buildKey-to build public key/private key

● Function-Description:

public boolean buildKey (String MerId, int KeyUsage, String KeyFile)

● Function:

To build public key/private key for generating signature or verifying digital signature

● Parameter Description:

String MerId: merchant ID，it is defined as an array of digits with the length of 15 and is assigned by ChinaPay

int KeyUsage: the way of using public/private key, fixed value is 0

String KeyFile: file pathe of public/private key(including file name) e.g. "d:\\MerPrk.key"

- Return Value:

  true: represents correct public/private key is found, and can use function of digital signature and digital signature verification

  false: represents failure of building public/private key, can't use function of digital signature and digital signature verification

- Note:

The method is in class chinapay. PrivateKey

2) Function signOrder-to generate digital signature

- Function-Description:

public String signOrder(String MerId, String OrdId, String TransAmt, String CuryId, String TransDate, String TransType)

- Function:

To generate digital signature for inputted parameters

- Parameter Description:

  String MerId – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.

  String OrdId – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

  String TransAmt– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

  String CuryId – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

  String TransDate– the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

  String TransType – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

- Return Value:

String CheckValue[256] --It is a digital signature string with the length of 256 bytes. It is generated using the input parameter values listed above.

- Note:

The method is in class chinapay. SecureLink

3) Function verifyTransResponse-to verify transaction response

- Function-Description:

  public boolean verifyTransResponse(String MerId, String OrdId, String TransAmt, String CuryId, String TransDate, String TransType, String OrderStatus, String CheckValue)

- Function:

To verify digital signature which inputs parameters

- Parameter Description:

String MerId – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.

String OrdId – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

String TransAmt– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

String CuryId – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

String TransDate– the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

String TransType – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

String OrderStatus– the status of the transaction. It is specified as an array of numbers sized4. See the "transaction response code table" for details.

String CheckValue - check value. It is the digital signature value with 256 bytes generated by   ChinaPay based on the transaction results.

- Return Value:

true : The value "0" means that the method has been executed successfully. That is to say, it is ChinaPay who sends the result to the merchant. And the merchant will be responsible for the subsequent processing. All the other return values indicate the failure of the function and the results can be ignored or simply put into the "garbage bin".

- Note:

This method is in class chinapay. SecureLink

4) Function Sign-to generate digital signature for a string
- Function-Description:

Public String Sign (String SignMsg)

- Function:

To generate digital signature for inputted string of parameters

- Parameter Description:

String SignMsg : is applied in the string for signature

- Return Value:

String CheckValue[256] -- It is a digital signature string with the length of 256 bytes. It is generated using the input parameter values listed above

5) Function verifyAuthToken- to verify signature for a string
- Function-Description:

public boolean verifyAuthToken (String PlainData, StringCheckValue)

- Function:

To verify the digital signature of target string

- Parameter Description:

  String PlainData- is applied in the string which generates digital signature

  String CheckValue- Verification value, to verify signature of a 256-byte string

- Return Value:

  True: represents success, otherwise is failure

3.5.2.3. Sample Codes

（the following code is not verified and only for reference）

- Sign Code

```
//initializing key files：

chinapay.PrivateKey key=new chinapay.PrivateKey();

chinapay.SecureLink t;

boolean flag;

String MerId, OrdId, TransAmt, CuryId, TransDate, TransType,ChkValue;

String plainData, ChkValue2 ;


flag=key.buildKey(MerId,0,"app/usr/chinapay/keys/MerPrk.key");

if (flag==false)

{

System.out.println("build key error!");

return;

}

t=new chinapay.SecureLink (key);

// sign to order

ChkValue= t.signOrder(MerId, OrdId, TransAmt, CuryId, TransDate, TransType) ;

// sign to a string

plainData = "test sign data ";

ChkValue2 = t.sign(plainData) ;
```

- Verify Signature Sample Code

```
chinapay.PrivateKey key=new chinapay.PrivateKey();

chinapay.SecureLink t;

boolean flag;

boolean flag1;

String MerId, OrdId, TransAmt, CuryId, TransDate, TransType,ChkValue;

String plainData, ChkValue2

flag=key.buildKey("999999999999999",0,"c:\\winnt\\PgPubk.key");

if (flag==false)

{

msg="build key error!";
```

14

```
    return;
    }
    t=new chinapay.SecureLink (key);
    flag1=t.verifyTransResponse(MerId,OrdId, TransAmt, CuryId, TransDate, TransType, OrderStatus, ChkValue);
// ChkValue returned by ChinaPay response.
    if(flag1!=0) {
            //handling the error of verifying signature.
    }


    // verifying the signature of a string
    plainData = "test sign data ";
    flag1 = t. verifyAuthToken (plainData, ChkValue2) ;    // ChkValue2 returned by ChinaPay response.
    if(flag1) {
      //handling the error of verifying signature.
        }
```

3.5.3. NetPayClient for Win32 Function Description
3.5.3.1. File list

Name	Path	Usage
netpay.dll	The same path as ChinaPay.dll , or copy the file to c:\winnt\system32(c:\windows\system for win95,98,etc)	To provide the function of order sign and ChinaPay's signature verification
netpay.lib	Unspecified (such as c:\netpay)	This file is required when invoking netpay.dll for compiling
MerPrk.key	Copy to the same path as windows c:\winnt for NT OR c:\windows for win95,98; or Unspecified, but needed to deploy function appointed file location and file name	Merchant's private key
PgPubk.key	Copy to the same path as windows c:\winnt for NT OR c:\windows for win95,98; or Unspecified, but needed to deploy function appointed file path and file name	ChinaPay's public key
ChinaPay.dll	Unspecified (such as c:\netpay)	Method package with netpay.dll in COM , need to register with regsvr32

Installation procedure

- Install automatically

Unzip the netpay package for NT platform and run NetPay4NTSetup.exe. The program will copy the files to the specified directory.

- Install manually

If there is exception thrown out during installation (probably due to the system abnormity), you may need to follow the procedures below to install the software manually.

- ・ Copy all the key files to the specified directory.
- ・ Register ChinaPay.dll by entering "regsvr32 [the path of gensign]" in the command line. This control can only work in a VC environment, it is suggested to use "Dependency Walker" to make sure that this DLL file has been set correctly.

3.5.3.2. Functions Description:

1) Set Private Key Path- setMerKeyFile
- Function Descripiton
  void setMerKeyFile (String KeyFile)

- Function
  Set private key path

- Parameters Descripiton:
String KeyFile:　file path of private key(including file name) e.g. "d:\\MerPrk.key

2) Unset private key path-unsetMerKeyFile
- Function Descripiton
  void unsetMerKeyFile ()

- Function
Unset previous private key path, recover default private key path

- Parameters Descripiton:
  None

3) Set ChinaPay Public Key Path-setPubKeyFile
- Function Descripiton
  void setPubKeyFile (String KeyFile)

- Function
  Set ChinaPay Public Key path

- Parameters Descripiton:
String KeyFile:　file path of private key(including file name) e.g.："d:\\ PgPubk.key"

4) Unset public key path- unsetPubKeyFile
- Function Descripiton

void unsetPubKeyFile ()

● Function

Unset previous ChinaPay public key path which setPubKeyFile sets, recover default public key path

● Parameters Descripiton:

None

5) Function sign-generate digital signature for order
● Function Descripiton

public String sign (String MerId, String OrdId, String TransAmt, String CuryId, String TransDate, String TransType)

● Parameters Descripiton:
  · String MerId – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.
  · String OrdId – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.
  · String TransAmt– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.
  · String CuryId – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.
  · String TransDate– the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.
  · String TransType – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

● Return values:

String CheckValue[256] -- It is a digital signature string with the length of 256 bytes. It is generated using the input parameter values listed above
● Note:

If COM is not called， load netpay.dll with LoadLibrary, need to find corresponding name with Dependency Walker.

6) Function check-to verify transaction response
● Function Descripiton

public String check(String MerId, String OrdId, String TransAmt, String CuryId, String TransDate, String TransType, String OrderStatus, String CheckValue)

- Parameters Descripiton:

String MerId – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.

String OrdId – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

String TransAmt– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

String CuryId – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

String TransDate– the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

String TransType – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

String OrderStatus– the status of the transaction. It is specified as an array of numbers sized 4. See the " transaction response code table" for details.

String CheckValue – check value. It is the digital signature value with 256 bytes generated by ChinaPay based on the transaction results.

- Return values:

true : The value "0" means that the method has been executed successfully. That is to say, it is ChinaPay who sends the result to the merchant. And the merchant will be responsible for the subsequent processing. All the other return values indicate the failure of the function and the results can be ignored or simply put into the "garbage bin".

- Note:

If COM is called, method return value is "0" in string—means signature verification successful.

7) **Function signData-to generate digital signature for a string**

- Function Descripiton

Public String Sign (String MerId, String SignMsg)

- Parameters Descripiton:

String MerId:　　merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay.

String SignMsg ：　is applied in the string for signature.

- Return values:

String CheckValue[256] -- It is a digital signature string with the length of 256 bytes. It

is generated using the input parameter values listed above

- Note:

  If COM is not called ， load netpay.dll with LoadLibrary, need to find corresponding name with Dependency Walker

8) Function checkData- to verify signature for a string
- Function Descripiton

  public String checkData (String PlainData, String CheckValue)

- Parameters Descripiton:

  String PlainData- is applied in the string which generates digital signature

  String CheckValue- Verification value, to verify signature of a 256-byte string

- Return values:

  "0": represents success, otherwise is failure

- Note:

  If COM is called, method return value is "0" in string—means signature verification successful.

3.5.3.3. Sample Codes

Take C# for example:

First, transform Com type information into .NET digits tlbimp ChinaPay.dll /out: ChinaPay_tsl.dll (tlbimp is the tool vs.net provides) in the project, if project->add reference， select com， input ChinaPay_tsl.dll

（the following code is not verified and only for reference）

- **Sign Sample Codes**

```
using System;
using ChinaPay_tsl;

namespace ConsoleApplication1
{
  class Class1
  {
    [STAThread]
    public static void Main()
    {
      //Sign
      NetPayClientClass a = new NetPayClientClass();
      //Set private key path
      a.setMerKeyFile("D:\\ MerPrK.key");  // if this method is not called, it will proceed
```

according to the default path

```
            String ChkValue;
            String MerId;   //merchant id


         //Sign order


         ChkValue=a.sign(MerId,"0000000000000001","000000001234","156","20070123","0
         001");
         // Sign a string
         String ChkValue2;
         String plainData =   "test sign data ";
         ChkValue2= a.signData(MerId,plainData);
       }
     }
}
```

- **Verify Signature Sample Codes**

```
using System;
using ChinaPay_tsl;

namespace ConsoleApplication1
{
  class Class1
  {
     [STAThread]
     public static void Main()
     {
       //verify signature
        NetPayClientClass a = new NetPayClientClass();
       //set private key path
        a.setPubKeyFile("D:\\ PgPubk.key");   // if this method is not called, it will proceed
                                         according to the default path
]        String ChkValue;
          String MerId; //merchant id
         //sign order
string                              flag                              =
a.check(MerId,"070699060500011","000000010000","156","20070615","0001","1001",
ChkValue); // ChkValue is return value which ChinaPay sends to merchant

// sign a string
String ChkValue2;
String plainData =   "test sign data ";
ChkValue2= a.signData(MerId,plainData);
```

```
          }
        }
      }
```

3.5.4. NetPayClient for C Function Description

3.5.4.1. File List

File name	File path	Usage
netpayclient.h	Unspecified	Applied in import file of C language etc
libnpc.so / libnpc.a	Unspecified	Method lib of signature data and signature verification (apply for .so or .a according to merchants' condition)
MerPrk.key	Set NPCDIR environment variable appointed file path, can't change file name or appoint NPCDIR environment variable, but need to call method appointed file path and name	Merchant signature private key
PgPubk.key	Set NPCDIR environment variable appointed file path, can't change file name or appoint NPCDIR environment variable, but need to call method appointed file path and name	ChinaPay signature public key

3.5.4.2. Functions Description

1) Set Private Key Path- setMerKeyFile
● Function Descripiton
void setMerKeyFile (char keyFile[256])

● Function
Set private key path.

● Parameters Descripiton:
char KeyFile: file path of private key(including file name) e.g. "d:\\MerPrk.key"

● Return values:

● Note:
2) Unset private key path-unsetMerKeyFile

- Function Descripiton
  Function Description: void unsetMerKeyFile ()

- Function
Unset previous private key path, recover default private key path
- Parameters Descripiton:
  None

3) Set ChinaPay Public Key Path-setPubKeyFile
- Function Descripiton
  void setPubKeyFile (char keyFile[256])

- Function
  Set ChinaPay Public Key path

- Parameters Descripiton:
  char keyFile: file path of private key(including file name) e.g.："d:\\ PgPubk.key"

4) Unset public key path- unsetPubKeyFile
- Function Descripiton
  void unsetPubKeyFile ()

- Function
  Unset previous ChinaPay public key path which setPubKeyFile sets, recover default public key path

- Parameters Descripiton:
- None

5) Generate digital signature Function – signOrder
- Function Descripiton
  int signOrder(char MerId[15], char OrdId[16], char TransAmt[12], char CuryId[3], char TransDate[8], char TransType[4], char CheckValue[256])

- Function
  To generate digital signature for inputted parameters

- Parameters Descripiton:
  char MerId[15] – merchant ID. It is defined as an array of digits with the length of 15. ChinaPay or the clearing bank assigns it to the authorized merchants.
  char OrdId[16] – order number. It is an array of numbers and has the length of 16, generated by the merchant's system and only the previously failed orders are allowed to be performed again.
  char TransAmt[12] – transaction amount. It is specified as an array of numbers

sized 12. For example, the value of "000000001234" represents 12.34RMB in this case.

char CuryId[3] – the type of the currency used. It is an array of numbers and has a size of 3. Currently, only the value of "156" is accepted to express RMB.

char TransDate[8] – the date of the transaction. This variable is defined as an array of numbers sized 8 and the format defined is YYYYMMDD.

char TransType[4] – the type of the transaction. It is a four-bytes long array and values between "0001" and "0002". "0001" represents payment transaction and "0002" refers to refund transaction.

char OrderStatus[4] – the status of the transaction. It is specified as an array of numbers sized 4. See the " transaction response code table" for details.

char CheckValue[256] – check value. It is the digital signature value generated by ChinaPay based on the transaction results.

- Return values:

The value "0" means that the method has been executed successfully. Otherwise means failure.

6) Verify transaction response –verifyTransResponse
- Function Descripiton

int verifyTransResponse(char MerId[15], char OrdId[16], char TransAmt[12], char CuryId[3], char TransDate[8], char TransType[4], char OrderStatus[4], char CheckValue[256])

- Function

To verify digital signature which inputs parameters

- Parameters Descripiton:

String MerId[15] – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.

String OrdId[16] – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

String TransAmt[12] – transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

String CuryId[3] – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

String TransDate[8] – the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

String TransType[4] – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

String OrderStatus[4] – the status of the transaction. It is specified as an array of

numbers sized4. See the "transaction response code table" for details.

String CheckValue[256] - check value. It is the digital signature value with 256 bytes generated by ChinaPay based on the transaction results.

- Return values:

The value "0" means that the method has been executed successfully. That is to say, it is ChinaPay who sends the result to the merchant. And the merchant will be responsible for the subsequent processing. All the other return values indicate the failure of the function and the results can be ignored or simply put into the "garbage bin".

- Note:
7) generate digital signature for a string – signData
- Function Descripiton

int signData (char MerId[15]，char *SignMsg，char ChkValue[256])

- Function

To generate digital signature for inputted string of parameters

- Parameters Descripiton:

char MerId[15] - merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay

char *SignMsg - is applied in the string for signature

char CheckValue[256] -It is a digital signature string with the length of 256 bytes. It is generated using the input parameter values listed above

- Return values:

0 means successful, otherwise is error codes.

8) Verify signature for a string –verifySignData
- Function Descripiton

int verifySignData (char * PlainData, char CheckValue[256])

- Function

To verify the digital signature of target string

- Parameters Descripiton:

char* PlainData - is applied in the string which generates digital signature

char CheckValue[256]- Verification value, to verify signature of a 256-byte string

- Return values:

The value "0" means that the method has been executed successfully. That is to say, it is ChinaPay who sends the result to the merchant. And the merchant will be responsible for the subsequent processing. All the other return values indicate

the failure of the function and the results can be ignored.

3.5.4.3. Sample Codes
- Sign Sample Codes

（the following code is not verified and only for reference）

```c
#include <stdio.h>
#include <stdlib.h>
#include "netpayclient.h"

int main (int argc, char* argv[])
{
    char MerId[16];
    char OrdId[17];
    char TransAmt[13];
    char CuryId[4];
    char TransDate[9];
    char TransType[5];
    char ChkValue[257];
    int flag;

setMerKeyFile("/app/netpay/key/MerPrk.key");    // if it is not set, files will be acquired
according to NPCDIR environment variables

// Sign an order
flag =signOrder(MerId, OrdId, TransAmt, CuryId, TransDate, TransType, ChkValue);
if(flag != 0) {
    //signature failure
}

// sign a string
plainData = "test sign data ";
flag = signData (MerId ,plainData, ChkValue);    // ChkValue is the signature of the string
if(flag != 0) {
    //signature failure
}
}
```

Verify Signature Sample Code

（the following code is not verified and only for reference）
```c
#include <stdio.h>
#include <stdlib.h>
#include "netpayclient.h"
```

```
int main (int argc, char* argv[])
{
     char MerId[16];
    char OrdId[17];
    char TransAmt[13];
    char CuryId[4];
    char TransDate[9];
    char TransType[5];
     char ChkValue[257];
     char OrderStatus[5];
     int flag;

setPubKeyFile ("/app/netpay/key/PubPrk.key"); // if it is not set, files will be acquired
according to NPCDIR environment variables

// receive fields that ChinaPay returns: MerId, OrdId, TransAmt, CuryId, TransDate,
TransType OrderStatus ,ChkValue

// sign the verified order

flag = verifyTransResponse(MerId, OrdId, TransAmt, CuryId, TransDate, TransType,
OrderStatus ,ChkValue);
if(flag != 0) {
    //signature failure
}

// sign a string
plainData = "test sign data ";
flag = verifySignData (plainData, ChkValue);  // ChkValue is the signature that ChinaPay
returns
if(flag != 0) {
    //signature failure
}
}
```

3.5.5. NetpayClient for C# Function Description
3.5.5.1. File List

File name	File path	Usage
netpay.dll	Installed in the corresponding path according to the need of project	To provide the method of deploying digital signature.
MerPrk.key	Unspecified saving path, but the function need to specify file path and file name.	Merchant's private key for signature.

PgPubk.key	Unspecified saving path, but the function need to specify file path and file name.	ChinaPay public key for signature.

- Saving the netpay.dll at specify path according to the need of project.
- In developing environment, by using"Adding Adduction" adds the netpay.dll into project, and could be called directly.

3.5.5.2. Functions Description
1) Build public key/private key object buildKey
- Function Descripiton
  public boolean builtKey (String MerId, int KeyUsage, String KeyFile)

- Function
  Used to build public key/private key object for generating signature or verifying digital signature.

- Parameters Descripiton:
  String MerId merchant ID，it is defined as an array of digits with the length of 15 and is assigned by ChinaPay.
  int KeyUsage the way of using public/private key, fixed value is 0.
  String KeyFile file pathe of public/private key(including file name) e.g. "d:\\MerPrk.key"

- Return values:
  true: represents correct public/private key is found, and can use function of digital signature and digital signature verification
  false: represents failure of building public/private key, can't use function of digital signature and digital signature verification

- Note:
  In netpay. NETPAY, this function can judge the status of current public key/private key by NetPay.NETPAY.PrivateKeyFlag or NetPay.NETPAY.PublicKeyFlag

2) Function signOrder – to generate digital signature
- Function Descripiton
  public String signOrder(String MerId, String OrdId, String TransAmt, String CuryId, String TransDate, String TransType)
  - Function
    To generate digital signature for inputted parameters.

  - Parameters Descripiton:
    String MerId – merchant ID, It is defined as a string with the length of 15 bytes which is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.

String OrdId – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

String TransAmt– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

String CuryId – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

String TransDate– the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

String TransType – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

- Return values:

String CheckValue[256] -- It is a digital signature string with the length of 256 bytes. It is generated using the input parameter values listed above

- Note:

The method is in class netpay. NETPAY

3) Function verifyTransResponse-to verify transaction response

- Function Descripiton

public boolean verifyTransResponse(String MerId, String OrdId, String TransAmt, String CuryId, String TransDate, String TransType, String OrderStatus, String CheckValue)

- Function

To verify digital signature which inputs parameters

- Parameters Descripiton:

String MerId – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.

String OrdId – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

String TransAmt– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

String CuryId – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

String TransDate– the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

String TransType – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and

"0002" refers to refund transaction.

String OrderStatus– the status of the transaction. It is specified as an array of numbers sized4. See the " transaction response code table" for details.

String CheckValue - check value. It is the digital signature value with 256 bytes generated by   ChinaPay based on the transaction results.

● Return values:

true : means that the method has been executed successfully. That is to say, it is ChinaPay who sends the result to the merchant. And the merchant will be responsible for the subsequent processing. All the other return values indicate the failure of the function and the results can be ignored or simply put into the "garbage bin".

● Note:

This method is in class netpay. NETPAY

4) Function Sign-to generate digital signature for a string
● Function Descripiton
Public String Sign (String SignMsg)

● Function
To generate digital signature for inputted string of parameters

● Parameters Descripiton:
String SignMsg : is applied in the string for signature

● Return values:
String CheckValue[256] -- It is a digital signature string with the length of 256 bytes. It is generated using the input parameter values listed above

5) Function verifyAuthToken- to verify signature for a string
● Function Descripiton
public boolean verifyAuthToken (String PlainData, StringCheckValue)

● Function
To verify the digital signature of target string

● Parameters Descripiton:
String PlainData- is applied in the string which generates digital signature
String CheckValue- Verification value, to verify signature of a 256-byte string

● Return values:
True: represents success, otherwise is failure

- Note:

This method is in class netpay. NETPAY

3.5.5.3. Sample Codes

(the following code is not verified and only for reference)

- **Sign Code**

```
string MerId, OrdId, TransAmt, CuryId, TransDate, TransType,ChkValue;
string plainData, ChkValue2 ;

//initialize key file
if (NetPay.NETPAY.buildKey("808080290000001", 0, "c:\\key\\MerPrk.key"))
{
     showInfo(">> set private key: succeed.\r\n\r\n");
}


// signature for the order
if (NetPay.NETPAY.PrivateKeyFlag)
{
     string ChkValue = NetPay.NETPAY. signOrder(MerId, OrdId, TransAmt, CuryId, TransDate,
TransType) ;
     showInfo(">>ostr: " + ostr + "\r\n\r\n");
     showInfo(">>ChkValue :" + ChkValue + "\r\n\r\n");
}


// signature for a string
string plainData = "80808029000000100000000102737650000000000001156201008060001";
if (NetPay.NETPAY.PrivateKeyFlag)
{
     string ChkValue2 = NetPay.NETPAY. Sign(plainData);
     showInfo(">>ostr :" + ostr + "\r\n\r\n");
     showInfo(">>ChkValue: " + ChkValue + "\r\n\r\n");
}
```

- **Verify Signature Sample Code**

```
String MerId, OrdId, TransAmt, CuryId, TransDate, TransType,ChkValue;
String plainData, ChkValue2
//initializing key file
if (NetPay.NETPAY.buildKey("999999999999999", 0, "c:\\key\\PgPubk.key"))
{
     showInfo(">>set public key:succeed.\r\n\r\n");
}
```

```
//verify signature of order
if (NetPay.NETPAY.PublicKeyFlag)
{
      bool flag= NetPay.NETPAY.verifyTransResponse(MerId,OrdId, TransAmt, CuryId, TransDate,
TransType, OrderStatus, ChkValue);   // ChkValue is returned by ChinaPay response.
      showInfo(">>ostr:" + ostr + "\r\n\r\n");
      showInfo(">>check:" + check + "\r\n\r\n");
      showInfo(">> verify signature:" + result + "\r\n\r\n");
      if(!flag) {
            // error of signature verification
      }
}


// verifying the signature of a string
plainData = "test sign data ";


if (NetPay.NETPAY.PublicKeyFlag)
{
      bool result = NetPay.NETPAY.verifyAuthToken(plainData, ChkValue2); // ChkValue2 returned
    by ChinaPay response
    showInfo(">>ostr:" + ostr + "\r\n\r\n");
    showInfo(">>check:" + check + "\r\n\r\n");
    showInfo(">>verifying signature: " + result + "\r\n\r\n");
    if(!flag) {
          // error of signature verification
    }
}
```

3.5.6.   NetpayClient for PHP Function Description
3.5.6.1. File List

Name	Path	Usage
netpayclinet.php	Installed in the corresponding path according to the need of user	To provide the method of deploying digital signature
MerPrk.key	Unspecified, but needed to deploy function appointed file path and file name	Merchant's private key
PgPubk.key	Unspecified, but needed to deploy function appointed file path and file name	ChinaPay's public key

3.5.6.2. Functions Description
1)   Function buildKey-to build public key/private key

31

- Function-Description:

  function buildKey ($keyfile)

- Function:

To build public key/private key for generating signature or verifying digital signature

- Parameter Description:

  $keyfile: file path of public/private key(including file name) e.g. "d:\\MerPrk.key",support relative path

- Return Value:

  true: return 15 bytes merchant id, else is false

2) Function signOrder-to generate digital signature

- Function-Description:

function signOrder($merid, $ordno, $amount, $curyid, $transdate, $transtype)

- Function:

To generate digital signature for inputted parameters

- Parameter Description:

  $merid – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay.

  $ordno – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

  $amount– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

  $curyid – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

  $transdate – the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

  $transtype –the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

- Return Value:

  NetPayClient generates a digital signature string with the length of 256 bytes using the input parameter values listed above. If the length of input parameter is invalid, the function return false.

3) Function verifyTransResponse-to verify transaction response

- Function-Description:

  function verifyTransResponse($merid, $ordid, $amount $curyid, $transdate, $transtype, $status, $checkvaluue)

- Function:

To verify digital signature is right or not

- Parameter Description:

  $merid – merchant ID, It is defined as a string with the length of 15 bytes that

is assigned by ChinaPay.

$ordid – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

$amount – transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

$curyid – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

$transdate – the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

$transtype – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

$status– the status of the transaction. It is specified as an array of numbers sized4. See the "transaction response code table" for details.

$checkvalue - check value. It is the digital signature value with 256 bytes generated by ChinaPay based on the transaction results.

● Return Value:

true means that the method has been executed successfully. That is to say, it is ChinaPay who sends the result to the merchant. And the merchant will be responsible for the subsequent processing. All the other return values indicate the failure of the function and the results can be ignored or simply put into the "garbage bin".

4) Function sign-to generate digital signature for a string
● Function-Description:
function sign ($msg)
● Function:
To generate digital signature for inputted string of parameters
● Parameter Description:
$msg : is applied in the string for signature

● Return Value:
NetPayClient generates a digital signature string with the length of 256 bytes using the input parameter values listed above.

5) Function verify- to verify signature for a string
● Function-Description:
function verify ($plain, $checkvalue)

● Function:
To verify the digital signature of target string

● Parameter Description:

$plain　　- is applied in the string which generates digital signature

$checkvalue　- Verification value, to verify signature of a 256-byte string

- Return Value:

True: represents success, otherwise is failure

3.5.6.3. Sample Codes

(the following code is not verified and only for reference)

- **Sign Code**

```php
<?php
include_once("netpayclient.php");

$merid = buildKey("MerPrk.key");
if(!$merid) {
        echo "导入私钥文件失败！";
        exit;
}

$ordid = "00" . date('YmdHis');
$transamt = padstr('1',12);
$curyid = "156";
$transdate = date('Ymd');
$transtype = "0001";
$version = "20070129";
$pagereturl = "$site_url/netpayclient_order_feedback.php";
$bgreturl = "$site_url/netpayclient_order_feedback.php";

$plain = $merid . $ordid . $transamt . $curyid . $transdate . $transtype . $priv1;
//sign for the order
$chkvalue = sign($merid, $ordid, $transamt, $curyid, $transdate, $transtype);
//sign for a string
$chkvalue = sign($plain);
if (!$chkvalue) {
        echo "签名失败！";
        exit;
}
?>
```

- **Verify Signature Sample Code**

```php
<?php
include_once("netpayclient.php");

$flag = buildKey("PgPubk.key");
if(!$flag) {
        echo "导入公钥文件失败！";
        exit;
```

```
        }

        $merid = $_REQUEST["merid"];

        $orderno = $_REQUEST["orderno"];

        $transdate = $_REQUEST["transdate"];

        $amount = $_REQUEST["amount"];

        $currencycode = $_REQUEST["currencycode"];

        $transtype = $_REQUEST["transtype"];

        $status = $_REQUEST["status"];

        $checkvalue = $_REQUEST["checkvalue"];

        $gateId = $_REQUEST["GateId"];

        $priv1 = $_REQUEST["Priv1"];


        $plain = $merid . $orderno . $amount . $currencycode . $transdate . $transtype . $status .
$checkvalue;


        // verify signature of order

        $flag = verifyTransResponse($merid, $orderno, $amount, $currencycode, $transdate, $transtype,
$status, $checkvalue);


        $flag   =   verify($plain, $checkvalue);

        if(!flag) {

            echo "<h2>验证签名失败！</h2>";

            exit;

        }

    ?>
```

4. Payment

4.1. Object

This chapter mainly instructs ChinaPay's merchants to connect with payment platform. In this way, merchants can develop their online payment transactions.

4.2. Transaction process

Card holders generate the order information from the merchant website, through the public payment gateway subsystem platform to do the payment process. The transaction processing includes order confirmation, payment processing, and payment complete three parts. More details as following:

【Step one】order confirmation

1. The card holder browse the website of merchant, chooses the payment options, generate the order information.

2. The card holder confirms order information, start payment process.

【Step two】payment processing

3. the card holder confirms the payment information, and the merchant website begin to applying the gateway for payment, then the gateway verify the validity of merchant and the integrality of message.

4. The payment gateway show the interface of payment channels, and card holder chose the payment channel.

5. Card holder input accounts and password and other verified information for safety on the channel he/she selected.

6. After the confirmation of card holder's safe information, begin the payment.

【Step three】payment complete
7. Payment channel return the payment result to the payment gateway.

8. Payment gateway shows the payment result to card holder, meanwhile informs merchant the payment result of website.

9. Website shows card holder the result of merchant.

10. The payment complete.

4.3. Connection Approaches

4.3.1. Connecting URL

Merchant can connect with ChinaPay's online payment system by the means of WEB service. URL of receiving transaction data is:

Test Environment：http://payment-test.chinapay.com/pay/TransGet

Production Environment：https://payment.chinapay.com/pay/TransGet

4.3.2. Version

Right now we have two kinds of payment versions:
● 20040916
● 20070129

Both of the versions submit the same fields, but different version may have difference, please pay attention to the explaining after the version.

4.3.2.1. Payment Version 20040916
● Content Submitted on Transaction Page

Merchant submits order information to ChinaPay's payment connection URL. Content of the FORM on transaction page should include(do remember to distinguish capital letters and small letters):

<form action="https://payment.ChinaPay.com/pay/TransGet" METHOD=POST> (the content of

action is the URL address to submit transaction data)

`<input type=hidden name="MerId" value="808080290000001"/>`((MerId is the Merchant ID assigned by ChinaPay, length of 15 bytes, necessary)

`<input type=hidden name="OrdId" value="0000000010096806"/>` (Order ID merchants submits to ChinaPay, length of 16 bytes, necessary)

`<input type=hidden name="TransAmt" value="000000001234"/>` (Order transaction amount, with 2 decimal,length of 12 bytes, add 0 from the left if inadequate, necessary)

`<input type=hidden name="CuryId" value="156"/>` (Order transaction currency, length of 3 bytes, only for Renminbi, necessary)

`<input type=hidden name="TransDate" value="20070801"/>`(Order transaction date, length of 8 bytes, necessary)

`<input type=hidden name="TransType" value="0001"/>` (Transaction type, length of 4 bytes, necessary)

`<input type=hidden name="Version" value="20040916"/>` (Payment connection version, necessary)

`<input type=hidden name="BgRetUrl" value="http://www.example.com/pay/Bgreturn.jsp"/>` (URL of receiving background transactions, within 80 byts, necessary)

`<input type=hidden name="PageRetUrl" value="www. example.com/pay/Pgreturn.jsp "/>` (URL of receiving page transaction, within 80 bytes, necessary)

`<input type=hidden name="GateId" value="">` (Payment gateway number, optional)

`<input type=hidden name="Priv1" value="Memo">`(Merchant private field, optional, within 60 bytes)

`<input type=hidden name="ChkValue" value="X…X">` (ASCII code of 256 bytes, necessary, digital signature of key data submitted by this transaction)

`</form>`

Explanation：

PageRetUrl: URL of receiving page transaction, used to lead the user back to merchant website pages after payment.

BgRetUrl: URL of receiving background transactions, used to record transaction and processing information, and it is invisible to user. Besides, ChinsaPay will judge whether need to retransmit the answer number in term of the http return code, in order to ensure the acceptance of background answer. The URL can not fill parameters. If you need to passing parameters, you can use" Priv1" field.

Priv1: Merchant private field, optional, ChinaPay fulfills the same content that merchants send to ChinaPay in this field, and returns to merchants.

GateId: Optional, means payment gateway number, if it is GateId（payment gateway number）, users directly enter payment webpage, if not, users enter gateway selection webpage.

OrdId: means order number, it can be defined by merchants, in payment version 20040916, if the merchant is secondary merchant whose merchant id's first six number is 808080, the order id must abide the rule as following:

The bytes of order id from 5 to 9 must keep the same with the byte of merchant id from 11 to 15.

- Digital Signature of Transaction Data

Digital signature is required when Merchant sends transaction data to ChinaPay. For Version 20040916, signature data should use "Function Sign - to generate digital signature for a string". Names of functions are:

NetPayClient for Java ： signOrder (refer to Chapter 3.5.2)

NetPayClient for Win32 ： sign (refer to Chapter 3.5.3)

NetPayClient for C ： signOrder (refer to Chapter 3.5.4)

NetPayClient for C# ： signOrder (refer to Chapter 3.5.5)

- Content of Receiving Transaction Data

ChinaPay will send transaction response to merchant when the payment has been finished. Page receiving URL and background receiving URL will both receive transaction data including the following fields. Take page FORM data for example.(pay attention to the capital letters and small letters)

<form name="SendToMer" method="post" action=""> （"action" here is the URL of submitting transaction data）

<input type="hidden" name="merid"   value="808080290000001"/>

<input type="hidden" name="orderno"   value="0000000010096806"/>

<input type="hidden" name="transdate"   value="20070801"/>

<input type="hidden" name="amount"   value="000000001234"/>

<input type="hidden" name="currencycode"   value="156"/>

<input type="hidden" name="transtype"   value="0001"/>

<input type="hidden" name="status"   value="1001"/>

<input type="hidden" name="checkvalue"   value=" X…X "/>

<input type="hidden" name="GateId"   value=" 0001"/>

<input type="hidden" name="Priv1"   value=" Memo"/>

</form>

- Description:

"status" represents the transaction statement, only when the value is "1001" means successful transaction, other values means failure, so after verify the sign date was send by ChinaPay, you still need to judge the statement value is "1001"

- Verifying Signature of Response Data

After payment transaction is finished, ChinaPay will send result information to merchant, which can be received by both page receiving URL and background receiving URL. Merchant should verify its signature to confirm whether this response is sent by ChinaPay. "Function check - to verify transaction response" should be used. Names of functions are:

NetPayClient for Java ： verifyTransResponse (refer to Chapter 3.5.2)

NetPayClient for Win32 ： check (refer to Chapter 3.5.3)

NetPayClient for C ： verifyTransResponse (refer to Chapter 3.5.4)

NetPayClient for C# ：verifyTransResponse (refer to Chapter 3.5.5)

4.3.2.2. Payment Version 20070129
- Content Submitted on Transaction Page

Merchant submits order information to ChinaPay's payment connection URL. Content of the FORM on transaction page should include(do remember to distinguish capital letters and small letters):

```
<form action="https://payment.ChinaPay.com/pay/TransGet" METHOD=POST>
```
("action" here is the URL of submitting transaction data)

```
<input type=hidden name="MerId" value="808080290000001"/>
```
(MerId is the Merchant ID assigned by ChinaPay, length of 15 bytes, necessary)

```
<input type=hidden name="OrdId" value="0000000000000006"/>
```
(Order ID merchants submits to ChinaPay, length of 16 bytes, necessary)

```
<input type=hidden name="TransAmt" value="000000001234"/>
```
(Order transaction amount, with 2 decimal, length of 12 bytes, add 0 from the left if inadequate, necessary)

```
<input type=hidden name="CuryId" value="156"/>
```
(Order transaction currency, length of 3 bytes, only for Renminbi, necessary)

```
<input type=hidden name="TransDate" value="20070801"/>
```
(Order transaction date, length of 8 bytes, necessary)

```
<input type=hidden name="TransType" value="0001"/>
```
(Transaction type, length of 4 bytes, necessary)

```
<input type=hidden name="Version" value="20070129"/>
```
(Payment connection version, necessary)

```
<input type=hidden name="BgRetUrl" value="http://www.example.com/pay/Bgreturn.jsp"/>
```
(URL of receiving background transactions, within 80 bytes, necessary)

```
<input type=hidden name="PageRetUrl" value="www.example.com/pay/Pgreturn.jsp "/>
```
)URL of receiving page transaction, within 80 bytes, necessary)

```
<input type=hidden name="GateId" value="">
```
（Payment gateway number, optional）

```
<input type=hidden name="Priv1" value="Memo">
```
(Merchant private field, within 60 bytes)

```
<input type=hidden name="ChkValue" value="X…X">
```
（ASCII code of 256 bytes, necessary, digital signature of key data submitted by this transaction）

```
</form>
```
- Descripiton:

PageRetUrl: URL of receiving page transaction, instructs users to return to merchants webpage after payment.

BgRetUrl: URL of receiving background transactions, for merchants to record transaction information and settlement, not visible for users. The URL can not fill parameters. If you need to carry parameters, you can use" Priv1" field. According to the return code of http, ChinaPay decides whether to repeat sending. Background responses reference number to assure receipt of background response.

Priv1: Merchant private field, join digital signature in payment version 20070129, ChinaPay fulfills the same content that merchants send to ChinaPay in this field, and returns to merchants.

GateId: Optional, means payment gateway number, if it is GateId（payment gateway number）,users directly enter payment webpage, if not, users enter gateway selection webpage.

OrdId: means order number, it can be defined by merchants.

● Digital Signature of Transaction Data

Digital signature is required when Merchant sends transaction data to ChinaPay. For Version 20070129, signature data should use "Function Sign-to generate digital signature for a string". Names of functions are:

NetPayClient for Java ：Sign (refer to Chapter 3.5.2)
NetPayClient for Win32 ：signData (refer to Chapter 3.5.3)
NetPayClient for C ：signData (refer to Chapter 3.5.4)
NetPayClient for C# ：Sign (refer to Chapter 3.5.5)

● Statement:

For Version 20070129, digital signature should use "Function Sign - to generate digital signature for a string", and the strings should be linked by the following order:

MerId	OrdId	TransAmt	CuryId	TransDate	TransType	Priv1

● Content of Receiving Transaction Data

ChinaPay will send transaction response to merchant when the payment has been finished. Page receiving URL and background receiving URL will both receive transaction data including the following fields. Take page FORM data for example.

<form name="SendToMer" method="post" action=""> （"action" here is the URL of submitting

transaction data）

<input type="hidden" name="merid"   value="808080290000001"/>

<input type="hidden" name="orderno"   value="0000000010096806"/>

<input type="hidden" name="transdate" value="20070801"/>

```
<input type="hidden" name="amount"       value="000000001234"/>
<input type="hidden" name="currencycode" value="156"/>
<input type="hidden" name="transtype"     value="0001"/>
<input type="hidden" name="status"        value="1001"/>
<input type="hidden" name="checkvalue" value=" X…X "/>
<input type="hidden" name="GateId"        value=" 0001"/>
<input type="hidden" name="Priv1"         value=" Memo"/>
```
</form>

- Statement:

"Status" represents transaction status. Only 1001 means success while other codes mean failure. Therefore, after verifying the signature data is sent by ChinaPay, merchant should judge whether the status code is 1001.

- Verifying Signature of Response Data

After payment transaction is finished, ChinaPay will send result information to merchant, which can be received by both page receiving URL and background receiving URL. Merchant should verify its signature to confirm whether this response is sent by ChinaPay. "Function check - to verify transaction response" should be used. Names of functions are:

NetPayClient for Java　：verifyTransResponse (refer to Chapter 3.5.2)
NetPayClient for Win32　：check (refer to Chapter 3.5.3)
NetPayClient for C　：verifyTransResponse (refer to Chapter 3.5.4)
NetPayClient for C#　：verifyTransResponse (refer to Chapter 3.5.5)

5. Refund

5.1. Object
Currently, ChinaPay employed plenty ways of refund, and this chapter mainly guide ChinaPay's merchant members access through single refund method to refund system. facilitate online payment business for success by ChinaPay refund transactions.

5.2. Transaction Process
After payment transition is finished, merchants can develop their refund transition payment service through payment gateway subsystem of payment service interface-NetPayClient, according to the order forms generated by the results of payment transition.

【Step 1】Refund Confirmation

1. Merchants according to the situation of payment transition select the refund program, then generate refund order or the information of refund bill.

2. Merchants confirm the information of refund ,and start refund transition.

【Step 2】Refund Response

3. Merchants confirm the data of refund transition, and apply refunding to ChinaPay with the data form regulated by ChinaPay. At the same time, payment gateway check the validity of merchants.

4. After ChinaPay received refund order, it will send back merchants the response of "Refund is accepted successfully." At the same time.

5. The response of success/failed refund will be send to notice merchants through the URL which is written by merchants at the time of submitting refund information.

【Step 3】Refund Dispose

6. ChinaPay completes refund transition with bank, which has manual intervention process ,and will generate refund information at last.

7. Merchants can look over the process of refund and the information of refund result from console platform.

【Step 4】Refund Finished

8. The refund operation is finished.

5.3. Connecting Approaches

5.3.1.　Connection URL

Merchant can connect with ChinaPay's enquiry system by the means of WEB service. URL of receiving transaction data is:

Test Environment: http://payment-test.chinapay.com/refund1/SingleRefund.jsp

Production Environment: http://console.chinapay.com/refund/SingleRefund.jsp

5.3.2.　Version

5.3.2.1.　Refund Version 20070129

● Content Submitted on Single Refund Page

Merchants submitted order information to ChinaPay's payment connection URL Content of the FORM on enquiry page should include:

<form action="https://bak.chinapay.com/refund/SingleRefund.jsp" METHOD=POST> ("action" here is the URL of submitting enquiry data)

<input type=hidden name="MerID " value="808080290000001"> (MerId is the Merchant ID assigned by ChinaPay, length of 15 bytes, necessary)

<input type=hidden name="TransType" value="0002"> (Transaction Type: 0002 represents refund transaction)

<input type=hidden name="OrderId" value="0000000010096806"> (Order ID, length of 16 bytes, necessary)

<input type=hidden name="RefundAmount " value="000000001234"> (Amount of merchant refund, with 2 decimal,length of 12 bytes, necessary)

<input type=hidden name="TransDate" value="20070801"> (Transaction date of the particular order, length of 8 bytes, necessary)

<input type=hidden name="Version" value="20070129"> (Version of single enquiry, necessary)

<input type=hidden name=" ReturnURL " value="http://www.example.com/ back1.jsp"> (URL of receiving refund status, optional, within 80 bytes)

<input type=hidden name="Priv1" value=" "> (Merchant's reserve field,within 40 bytes, necessary, optional)

<input type=hidden name="ChkValue" value="X...X"> (ASCII code of 256 bytes, necessary)

</form>

· Statement:

TransType is transaction type, 0002 represents refund. ReturnURL is refund respond, whose content is optional. If merchants would like to get informed when the refund status changed, the files should be filled in.

Priv1 is merchant's private field, necessary, whose data is defined by merchant but should not be repeated. In order to avoid submitting refund request for several times, data in this field will be verified if it is repeated. If the refund request has been submitted before, it is considered that it has been received and will not be processed again

● Digital Signature of Transaction Data

Digital signature is required when Merchant sends transaction data to ChinaPay. For Version 20070129, signature data should use "Function Sign - to generate digital signature for a string". Names of functions are:

NetPayClient for Java　：Sign (refer to Chapter 3.5.2)
NetPayClient for Win32　：signData (refer to Chapter 3.5.3)
NetPayClient for C　：signData　(refer to Chapter 3.5.4)
NetPayClient for C#　：Sign (refer to Chapter 3.5.5)

· Statement

For Version 20070129, signature data uses "Function Sign - to generate digital signature for a string", so merchants should contact stings follow some sequence ,the sequence is as follows:

MerID	TransDate	TransType	OrderId	RefundAmount	Priv1

● Content of Refund Response Data

ChinaPay's server program will send back "refund request received" response to merchant after processing refund request. When the status of particular refund changes into success or failure, corresponding response will also be sent to merchant.

· Format of "success" response:

ResponseCode=value0&MerID=value1&ProcessDate=value2&SendTime=value9&TransType=value3&OrderId=value4&RefundAmout=value5&Status=value6&Priv1=value7&CheckValue=value8

· Format of "failure" response:

ResponseCode=value0&Message=message_string

· Statement:

Merchant can judge whether the refund is successful via ResponseCode. ResponseCode value of 0 represents successful refund while other Response codes mean failed refund.

· Definition of each field in "success" response:

ResponseCode : The code of response, the common value is 0(When the value is 0, it will has other field data sent back to check the signature of the message).

MerID： Merchant ID, 15 bytes figure

ProcessDate: ChinaPay's process date, 8 bytes figure

SendTime： Time of ChinaPay sending response code, 6 bytes figure(hhmmss), this field is not in digital signature

TransType： Transaction type, 4 bytes figure

OrderId： Original order ID, 16 bytes figure

RefundAmout： Refund amount, 12 bytes figure

Status： Status of particular refund, 1 byte figure

1	Refund request submitted successfully
3	Refund is succeed
8	Refund is fail

Priv1: Merchant's private field, maximum 40 bytes

Checkvalue: Signature verification, 256 bytes

Definition of each field in "failure" response:

ResponseCode : Response code, 3 bytes figure

Message: Explanation of response code

● Signature Verification of Response Data

Response data will be sent to merchant when the refund request is successfully submitted or the refund status changes into success or failure. After receiving response data, merchant should verify the signature to confirm whether this data is sent by ChinaPay, using "Function check - to verify transaction response". Names of functions are:

NetPayClient for Java ： verifyAuthToken (refer to Chapter 3.5.2)

NetPayClient for Win32 ： checkData (refer to Chapter 3.5.3)

NetPayClient for C ： verifySignData (refer to Chapter 3.5.4)

NetPayClient for C# ： verifyAuthToken (refer to Chapter 3.5.5)

● Statement:

For Version 20070129, verifying digital signature of refund transaction should use "Function check - to verify transaction response", and the strings should be linked by the following order:

MerID	ProcessDate	TransType	OrderId	RefundAmount	Status	Priv1

6. Query

6.1. Object

Every transaction submitted to ChinaPay can be inquired both on merchant console, and inquiry service. This chapter conducts merchant to set up connection with ChinaPay's enquiry system, so that they can facilitate merchant to inquire transactions.

6.2. Transaction Process

Merchant connect inquiry order information that meet merchant console according informative details of payment transition, then, they undertake the inquiry operation of transactional   details. The processing flow of inquiry includes

【Step 1】Inquiry Confirmation

1.   Merchant generates inquiry order information.

2.   Merchant confirms inquiry order information, which should meet ChinaPay's code requirement.

【Step 2】Inquiry Action

3.   After confirming inquiry order, merchant applies to ChinaPay for inquiring transactional details with the data format regulated by ChinaPay.

4.   ChinaPay deals with the inquiry of transactional details according to merchant's information and inquiry order information submitted.

【Step 3】Inquiry Finished

5.   ChinaPay's server program send back the response of query results to merchant once it finished undertaking the request of querying transactional details.

6.   Transactional details' query finished

6.3. Connection Approaches

6.3.1.   Connection URL

Merchant can connect with ChinaPay's single enquiry system by the means of WEB service. URL of receiving transaction data is:

Test Environment: http://payment-test.chinapay.com/QueryWeb/processQuery.jsp

Production Environment: http://console.chinapay.com/QueryWeb/processQuery.jsp

In addition, as we add extra note that ChinaPay adopts the way of flow control to permit merchant accessing to, following conditions should be satisfied:

- Request will be responded only if it is submitted from IP address designated by merchant that has the right to inquire transaction, and the interval between two successful enquiries should be longer than the system's set interval. Otherwise system will report "illegal merchant request" (Error code 111).
- Transactions of failure status which are inquired in a fix time should not exceed some volume, otherwise system will report "over-flow control" (Error code 305). Therefore, when merchant applies for single enquiry function, its server's IP

address and Merchant ID should be provided.

6.3.2. Version

6.3.2.1. Enquiry Version 20060831(Single Query)

- Content Submitted on Single Enquiry Page

Merchant submits order information to ChinaPay's payment connection URL. Content of the FORM on enquiry page should include:

<form action="http://console.chinapay.com/QueryWeb/processQuery.jsp" METHOD=POST>

("action" here is the URL of submitting enquiry data)

<input type=hidden name="MerId" value="808080290000001"> (MerId is the Merchant ID

assigned by ChinaPay, length of 15 bytes, necessary)

<input type=hidden name="TransType" value="0001"> (Transaction Type: 0001 represents consumption transaction)

<input type=hidden name="OrdId" value="0000000010096806"> (Order ID, length of 16 bytes, necessary)

<input type=hidden name="TransDate" value="20070801"> (Transaction date of the particular order, length of 8 bytes, necessary)

<input type=hidden name="Version" value="20060831"> (Version of single enquiry, necessary)

<input type=hidden name="Resv" value=" "> (Merchant's reserve field)

<input type=hidden name="ChkValue" value="X…X"> (ASCII code of 256 bytes, necessary)

</form>

- Digital Signature of Transaction Data

Digital signature is required when merchant sends transaction data to ChinaPay. For Version 20060831, signature data should use "Function Sign - to generate digital signature for a string". Names of functions are:

NetPayClient for Java	：	Sign (refer to Chapter3.5.2)
NetPayClient for Win32	：	signData (refer to Chapter 3.5.3)
NetPayClient for C	：	signData (refer to Chapter 3.5.4)
NetPayClient for C#	：	Sign (refer to Chapter 3.5.5)

- Statement:

For Version 20060831, digital signature of enquiry transaction should use "Function Sign - to generate digital signature for a string", and the strings should be linked by the following sequence:

MerID	TransDate	OrdId	TransType

- Content of Query Response Data

ChinaPay's server program will send back the enquiry result to merchant after undertaking query request.

Format of "success" response(Note case sensitive):

> ResponeseCode=value1&merid=value2&orderno=value3&amount=value4&currencyc
> ode=value5&transdate=value6&transtype=value7&status=value8&checkvalue=value9
> &GateId= value10&Priv1= value11

Format of "failure" response(Note case sensitive):

> ResponeseCode=value0&Message=message_string

- Statement:

Merchant can judge whether the query is successful via ResponseCode. ResponseCode value of 0 represents successful query while other Response codes mean failed query.

Definition of each field in "success" response:

ResponeseCode: Response code, successful value is 0

merid：   Merchant ID, 15 bytes figure

orderno：   Enquiry Order ID, 16 bytes figure

amount：   Transaction amount, 12 bytes figure

currencycode：   Transaction currency, 3 bytes figure

transdate: Transaction date, 8 bytes figure

transtype：   Transaction type, 4 bytes figure

status：   Transaction status, 4 bytes figure

checkvalue：   Digital signature, 256 bytes

GateId：   Transaction gateway ID, 4 bytes

Priv1：   Merchant private field

Definition of each field in "failure" response:

ResponeseCode: Response code, 3 bytes figure

Message：   Explanation of response code

- Signature Verification of Response Data

ChinaPay will send the enquiry information to merchant when completing enquiry transaction, using "Function check - to verify transaction response". Names of functions are:

NetPayClient for Java ：verifyTransResponse (refer to Chapter 3.5.2)

NetPayClient for Win32 ：check (refer to Chapter 3.5.3)

NetPayClient for C ：verifyTransResponse (refer to Chapter 3.5.4)

NetPayClient for C# ：verifyTransResponse (refer to Chapter 3.5.5)

7. **Appendix A Common Connection Problems and Solutions**

1)   In the development process , reported 157 error code.

QA: Check to see if GATEID is 0001, if it is, delete 0001

2)   After payment online, how to do, if click submit button, the page presents " The standard data field cannot be blank"?

QA: Merchant checks whether every piece of program is written according to

<<NetPayClient_MerchantOperationManual>>. When come up with such report, it expresses program designer doesn't meet with the manual, there are some stupid mistakes such as spelling mistake, forgetting to put in important data and so on. Remind merchant to check the parameters as follows: MerId, OrdId, TransAmt, CuryId, TransDate, TransType and some relative programs.

Typical case:

One company's program's designer input:

……

<input type=hidden name="cruyId" value="156">

……

The correct one is:

<input type=hidden name="curyId" value="156">. A word's error contributes to the difficult of data reading. That is, function can't find the important information included by curyId, which contributes to the report of "Standard data field can not be blank!"

3)　How to deal with the problem that the system report 500 error code during merchant's testing process?

QA: This error is systematic error, which belongs to the error of compiling program.

That is because merchant mixed test submit URL and official submit URL.

Typical case:

When compile programs , designer input:

……

<form action=https://payment.chinapay.com/pay/transget method=post>

……

This is test merchant, so the right content after action is http://payment-test.chinapay.com/, that is to say, the right input is:

<form action=http://payment-test.chinapay.com/pay/transget method=post>。

4)　How to deal with the problem that merchant comes up with -109 error code?

QA:-109 error code represents encryption is not successful, which may be caused by many problems. Currently, the main reason is that the signature method merchant used is not right, which result in fail signature. We should remind merchant to watch the version of interface, Version 20070129 is different from version 20040916 in technique developing. Wrong form of signature may cause fail signature. Remind merchant to read the manual from p21 to p25 carefully.

V07 should realize via signing on a bunch of strings. The detail differentiation is as follows:

	InterfaceType	20040916	20070129
Sending Transaction	NetPayClient for Java	signOrder	sign
	NetPayClient for Win32	sign	signData
	NetPayClient for C	signOrder	signData

	NetPayClient for C#	signOrder	Sign
Receiving	NetPayClient for Java	verifyTransResponse	verifyTransResponse
Transaction	NetPayClient for Win32	check	check
	NetPayClient for C	verifyTransResponse	verifyTransResponse
	NetPayClient for C#	verifyTransResponse	verifyTransResponse

When comes to such error, you can utilize the method referred in 2.1, check the sending code for client.

5) How to deal with the problem that merchant comes to illegal length?

QA: Illegal length is because of merchant's wrong configuration that the length of parameters is not follow actual demands. For example ,OrderId is not 16 bytes,TransDate is not 8 bytes, TransAmt is not 12 bytes, Checkvalue is not generated, and so on, all of which may cause illegal length.

The detail statement of sending parameters is as follows:

String MerId – merchant ID, It is defined as a string with the length of 15 bytes that is assigned by ChinaPay or the clearing bank when the merchant agrees to use the service provided by ChinaPay.

String OrdId – order number. It is specified as a 16-byte long numeric string that is generated by the merchant's system, and previously failed transactions could be paid again.

String TransAmt– transaction amount. This variable is defined as a numeric string and has the length of 12. For example, the string "000000001234" represents 12.34RMB in this case.

String CuryId – the type of the currency used. It is a string containing 3 bytes. Currently, only the value "156" is accepted to represent RMB in the system.

String TransDate– the date of the transaction. This numeric string is defined with the length of 8 to represent the transaction date with the format of YYYYMMDD.

String TransType – the transaction type. It is a string sized 4 and values between "0001" and "0002". Hereinto, "0001" represents payment transaction and "0002" refers to refund transaction.

Return Value

String CheckValue - check value. It is the digital signature value with 256 bytes generated by ChinaPay based on the transaction results.

6) What's the problem that ERRORCODE=110?

QA: Background return URL or Page return URL is too long. The length of Background return URL can't over 80 bytes. If it is over 80 bytes ,the system will report 110 error code. Advice merchant alter Background return URL and Page return URL, limit the length within 80 bytes.

7) How to deal with the problem that AXTIVEX control can't create object or DLL is

not registered during the development process of ASP?

QA:

a)   Let merchant to check whether the SCRIPT of system XP has been updated to the latest version.

b)   Let merchant to check whether lode CHINAPAY.DLL and CPNPC.DLL correctly, and put NETPAY.DLL on the SYSTEM32 folder.

8)   What's the problem that system reports illegal page in the development process?

QA: Submit page should use popup windows. If merchant use frame interface, that the submission page will not redirect properly, that would report illegal page. Advice merchant to alter the structure of website, and to use popup windows instead of directing in the frame of page.

9)   When sending transaction, some other error report information

status code	Transaction Description
1001	Purchase transaction succeeded
1003	Refund transaction succeeded
1005	Refund order cancel succeeded
others	All the other response codes refer to a failed transaction. Please check the detailed description from ChinaPay's Merchant Management System or error information files of banks

10) During the system testing, successful payment but occur verifying signature mistake when return to merchant website page.

QA: firstly, the program developing must keep accord to the content of << NetPayClient User Manual>>. Meanwhile, you should care when accepting the response, the signature method of version 20070129 and version 20040916 are identical. Please pay especial attention to this.

Typical Example:

One company's programmer input following content when testing company:

……

Chkvalue = request.getparameter("chkvalue")

……

Because the sending parameter's name is"ChkValue", when feedback fetch the "ChkValue" too. In fact, this parameter should be "checkvalue", so the transmit parameter value is null, and results to failure of verifying signature. That is to say:

Chkvalue = request.getparameter("checkvalue")

This mistake as a result of the merchant didn't code the program according to << NetPayClient User Manual >>.

11) What should we do when the failure of initializing public key occurs?

QA: If merchants develop their program in JAVA, please check that whether merchants have input the public key number instead of merchant ID.

Typical Example:

One company's programmer input following content when testing company:

……

Flag=key.buildKey("808080002100999",0"c:\winnt\\PgPubk.key");

……

But, the correct code should like this:

……

Flag=key.buildKey("999999999999999",0"c:\winnt\\PgPubk.key");

……

12) What we should do when the -118 mistake occurs?

QA: Generally speaking, this mistake does not occur frequently. It is mainly because the merchant use the incorrect private key or transmit the wrong parameters to verifying signature method which lead to verify signature error. For example, the testing merchant use the producing merchant's private key or signature method lead to incorrectly parameter transmitting and verifying signature failure.

Typical Example:

One merchant use producing public key when test testing online system, result to -118 mistake, after modified to testing merchant public key the error removed.

13) What about can't receive the background answer?

QA: Checking that whether the response acceptance code have something wrong with the judgment which responsible for judging the qualification of accepting response; and ensure the correct accepting response URL when sending transaction.

14) How to inform ChinaPay when accept background response successfully?

QA: WEBSEVER feedback automatically, generally, successfully acceptance will return number 200 automatically, so merchant don't need any parameter. Only when the value of OrderStatus is "1001" means successful transaction, the rest values all mean failing transaction. And merchant need to judge this then adds to own database.

15) What about the failure of loadlib(CHINAPAY.DLL)?

QA: Netpay.dll must be put in the system32 folder.

8. Appendix B Response Code Description

8.1. ChinaPay transaction status code

Form 3 ChinaPay transaction status code

status code	Transaction Description
1001	Purchase transaction succeeded
1003	Refund transaction succeeded
1005	Refund order cancel succeeded
others	All the other response codes refer to a failed transaction. Please check the detailed description from ChinaPay's Merchant Management System or error

information files of banks

8.2. NetPayClient API Error Codes

Form 4 NetPayClient API Error Codes

status code	Status Information
All methods	
-111	Private key path is not set, or NPCDIR environment variable is not set
Signature methods	
-100	NPCDIR environment variable is not set
-101	Merchant private key file does not exist or can't be opened
-102	private key file format error
-103	private key merchant id is not compatible with signature merchant id
-130	Length of string for signature is blank
Signature verification methods	
-112	ChinaPay public key file does not exist or can't be opened
-113	Public key file format error
-114	Public key file error
-118	Signature verification failure
-134	Public key file path set error

8.3. Inquiry，Single Refund Error Codes

ERROR CODE	ERROR MESSAGE
101	Merchant id error or is blank
102	Order number of transaction response inquiry is blank
103	Transaction date is blank
104	Request transaction type error
105	Merchant id length error
106	Order number length error
107	Transaction date length error
108	Transaction type is blank or length error
109	Version number is blank
110	Version number error
111	Merchant request forbidden
112	Transaction type error
116	Amount is blank
117	Merchant private field length error

118	Signature field error
121	Refund amount does not equal to original order amount
122	Refund amount should be less than original order amount
123	Total refund amount is more than original order amount
201	Plaintext data is blank
201	Signature data does not exist
202	Signature error
203	Signature verification failure
204	Signature verification data error
205	Repeat submitting the transaction
301	Inquired transaction does not exist
302	Inquiry database error
303	Response data package error
304	Response data string transformation error
305	Exceed volume control range
307	Corresponding data is not found
404	Other internal errors

ChinaPay E-Payment Service Co.,Ltd

http://www.ChinaPay.com